TOWARDS UNSUPERVISED LEARNING FOR AUTOMATIC MULTI-CLASS OBJECT DETECTION IN SURVEILLANCE VIDEOS

Hasan Celik, Alan Hanjalic, Emile A. Hendriks

Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
h.celik@tudelft.nl

ABSTRACT

Object detection is a critical step in automated surveillance. A common approach to constructing object detectors consists of annotating large datasets and using them to train the detectors. However, due to inevitable limitations of a typical training data set, such supervised approach is unsuitable for building generic surveillance systems applicable to a wide variety of scenes and camera setups. In our previous work we proposed an unsupervised method for learning and detecting the dominant object class in a general dynamic scene observed by a static camera. In this paper, we investigate the possibilities to expand the applicability of this method to the problem of multiple dominant object classes. We propose an idea how to approach this expansion, and perform a proof-of-concept evaluation of this idea using a representative surveillance video sequence.

Index Terms— Object detection, Surveillance, Pattern classification, Clustering, Unsupervised learning

1. INTRODUCTION

Progresses in digital capture, storage and computing power in recent years have made smart automated monitoring and surveillance systems feasible. Consequently, new challenges arose in the computer vision and machine learning fields concerning the design of efficient and solid algorithms for tasks such as object detection, recognition and tracking. Our emphasis in this paper is object detection, which is the essential component of automated surveillance and monitoring solutions.

The problem of object detection has typically been addressed through supervised, offline learning approaches, which resulted in an abundance of solutions for various object classes, scenarios and applications (e.g. [3, 4]). These existing detectors detect different object classes generally one at a time, and were proved effective in numerous cases. However, their performance is limited by the quality of the training data, which can be expressed by the extent to which this data covers the whole scope of deviations in object appearance specific for a given application context. These deviations typically result from occlusions and the changes in view point and lighting conditions. Compared to this, the unsupervised methods do not rely that heavily on the domain knowledge and could be able to adjust automatically to the peculiarities of the observed scene. One such method aiming at learning simultaneously two classes of objects dominating the scene is introduced and evaluated in this paper.

2. RELATED WORK AND CONTRIBUTION

Levin et al. [2] developed a system learned in a semi-supervised fashion using co-training. Two distinct detectors are used to train each other and improve the performance of the system. Likewise, Javed et al. [8] used co-training to improve the performance of an initial classifier by selecting new training examples using PCA. Still, both systems necessitate a non-negligible amount of supervision for labeling during initialization. In [1], Nair et al. presented a framework that uses a rough detector to collect training samples to train a pedestrian detector. Yet their rough detector is simply a manually pre-defined object size ratio. More recently, Wu et al. presented in [5] an approach to online (re)-training of a detector based on the outputs of an oracle (coarse detector) using boosting. As boosting focuses on difficult examples during training, this may cause instability if some examples are wrongly labeled. Furthermore, like [2], the method [5] also needs supervision for the initial stages, and it can only learn objects having the appearance very similar to the original samples. Nearly all mentioned approaches have been applied in one context only, e.g. for pedestrian or car detection, with the exception of Javed et al [8], where the same procedure applies for both pedestrians and cars. Compared to these methods, we presented in [9] a detector of dominant objects able to calibrate itself in an autonomous and robust fashion. This approach is online and unsupervised: it automatically obtains training samples from the input scene.

All the approaches mentioned above deal with only one class of objects at a time. In surveillance scenarios, however, multiple object classes appear often at the same time, which requires simultaneous learning of multiple object detectors.

3. THE PROPOSED APPROACH

In our previous work [9] we demonstrated the effectiveness of the unsupervised framework for learning the model of a dominant object class in a given surveillance video sequence and detecting all further instances of this object class in that sequence. As illustrated in Figure 1, the first component of this framework is a *Coarse Object Detector* (COD). The COD extracts information from the video of the observed scene and uses it to identify potential training examples of the dominant object class. In its advanced form, the framework may include a *Clustering* module, where the different training examples corresponding to different object classes are filtered and grouped together to provide reliable samples for training of the *Fine Object Detector* (FOD). While the COD uses simple object features to find candidate samples of object classes, the Clustering module is much more sophisticated and serves to refine the COD output by grouping the "good" candidates into one cluster and "bad" into the other. The good cluster is further used as a positive training set for the FOD. The negative set for training the FOD is obtained by randomly cropping image parts from the video in frames containing no motion, at different locations and scales. The trained FOD is then used to detect all further instances (also the stationary ones) of the dominant object class in new frames of the observed scene.

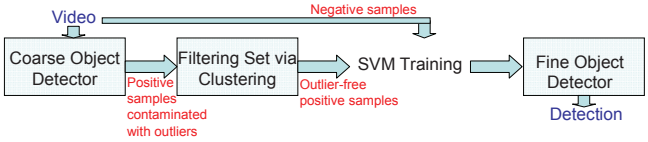

Figure 1: Block diagram of the framework [9] to train a dominant object detector in an unsupervised fashion.

The framework in Figure 1 is applicable in many surveillance scenarios, where a single dominant object class can be identified. Typical examples of such a class are "people" in a shopping mall or at a railway station, or "cars" on a highway. However, the situations involving multiple dominant object classes are also frequent, as illustrated by the real world scene example in Figure 2 where both "cars" and "people" dominate, i.e. both are statistically relevant.

Figure 2: Scene involving two dominant object classes (pedestrians and cars).

In the remainder of this section we investigate to which extent it would be possible to expand the framework realization from Figure 1 into a multi-class object detection problem. We do this by modifying the post-processing (clustering) step following the COD in Figure 1. Instead of focusing on one "good" cluster only, we now target multiple "good" clusters of COD outputs. Subsequently, each of these clusters can be used to train a specific object detector aimed at finding new instances of their corresponding elements. Such a solution is illustrated in Figure 3.

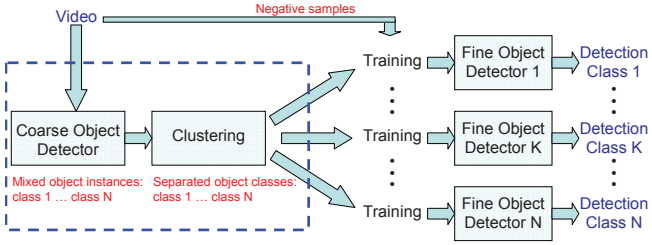

Figure 3: Block diagram of a framework to simultaneously train multiple dominant object detectors in an unsupervised fashion.

3.1. Extension from the single object class case towards the multi-class case

In [9], in order to optimize the COD output, we first collected the long-term statistics of the moving blobs corresponding to the dominant object class and modeled the perspective deformation of one of its simple dimensional features. Using this perspective model, further instances of the "proper" moving blobs are extracted. Figure 4 shows the result of applying this COD concept to the sequence illustrated in Figure 2. As in this sequence two dominant object classes appear, namely the pedestrians and the cars, we expected that the difference in the dimensional feature values obtained for the blobs of different object classes would result in two perspective models, the outputs of which could be fed into the corresponding clustering modules. We noticed, however, that still one perspective model is generated that results in mixed object samples as shown in Figure 4. The reason for this bad COD output can clearly be searched in the fact that the dimensional feature used (blob height in this case) has values that are close for both cars and people.

Figure 4: Output of the COD on the sequence in Figure 2.

An intuitive approach to compensate for the above could be to use a more sophisticated dimensional feature. Following this approach moves the complexity from the

Clustering module to the COD, which – if done to the extreme – would even eliminate the need for the refinement of the COD results in the Clustering module and therefore would make the Clustering module obsolete. However, the drawback of introducing more sophistication in selecting the dimensional features is inevitably the tuning of this feature for a particular object class, which would reduce the generic applicability of our framework.

Figure 5: Image patches corresponding to blobs appearing in the scene of Figure 2.

We therefore propose an alternative approach to solving the multi-class object detection problem, in which we minimize the complexity of the COD and maximally rely on the clustering module that we already defined robustly in [9] and apply now to a more diverse set of candidate object samples. In our new approach the COD reduces to the detector of moving blobs (e.g. by background subtraction). The blobs are rescaled to a size for a given ordinate using a perspective distortion model [12]. The rescaling enables a fair comparison between objects appearing far in the background and the ones close to the camera. Examples of such blobs and the corresponding objects from the scene in Figure 2 are given in Figure 5. In the next step, these blobs are submitted as inputs into the clustering algorithm that we explain in more detail in the following subsection.

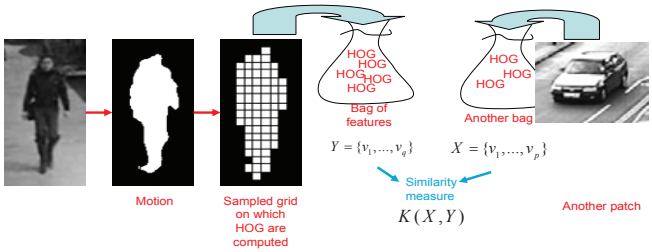

Figure 6: Positive patch, corresponding motion and blocks on which HOG features are computed. The set of features per patch can be considered as a bag of features, with different number of elements in it.

3.2. Separation into multiple object classes

3.2.1. Features

We base the clustering of the COD outputs on the HOG features described in [3]. Being conceptually similar to SIFTs [6], HOGs are densely sampled histograms of oriented gradients. They are computed on blocks of size 8×8 as illustrated in Figure 6, and are represented by a vector corresponding to frequencies of orientations of the gradient in that particular block. Originally, Dalal et al. [3] concatenate these blocks over a 4×4 block neighborhood. We could also adopt this grouping, but this would imply that the totality of blocks would be used to characterize a given patch. The disadvantage would be the inclusion of background parts to be taken into account. We resolve this issue by only considering the blocks of the patches which correspond to motion.

3.2.2. Similarity measure

We can represent each extracted patch as a collection x of n-dimensional HOG vectors

$$x = \{v_1, v_2, ..., v_i, .., v_p\}$$

where $v_i \in \mathbb{R}^n$ and p is the number of these vectors. The similarity measure $K(x,y)$ between two sets (bags of HOGs) x and y is defined in [7] using the Bhattacharya kernel between the two bags of features. The bags of features are fitted with Gaussian distributions after being mapped in the Reproducing Kernel Hilbert Space. Then, the problem of measuring the distance between two bags of features is transformed into a problem of measuring the distance between two distributions. Note that x and y do not necessarily have the same number of vectors, which motivates our choice for a kernel function between distributions estimated from samples [7]. The similarity is computed for all (x,y) pairs which gives a matrix $S \in \mathbb{R}^{m \times m}$ where m is the total number of elements to be clustered. To measure this similarity, we consider the HOGs of the parts which correspond to motion. The motion produced by an object is indeed a characteristic feature of its shape. In the case of an object from another class, the shape is different.

3.2.3. Clustering

A similarity-based clustering method is used to split the COD output set into multiple clusters. Multiple variants of clustering based on similarities are available in the free *PRTools* package [11]. Concerning the number of clusters, it has to be deduced from the spectrum of the similarity matrix, or if not obvious from this spectrum, defined manually.

3.3. Fine Object Detection

As in our previous approach [9], we selected the detector presented in [3]. It uses SVM on densely sampled HOG features. Initially used for pedestrian detection, we demonstrated its effectiveness on other classes such as cars.

4. RESULTS

Our initial experiments were conducted on the patches extracted from a single video (cf. Figure 2). This is a publicly available video used for performance evaluation of surveillance-related algorithms [10]. Two videos of the same scene at different moments are available. We used one of them for training (13,167 frames) and the other for testing (3,929 frames), which results in 17,096 frames for the experiments. The scene involved in this video contains pedestrians and cars as dominant object classes.

In total, via the COD, we extracted 689 potential training example candidates from the sequence. Via annotation, manually achieved for evaluation only, it appears that 543 of these correspond to single or multiple pedestrians, 119 to single or multiple cars, and the remaining 27 are either a mix of both or other objects (e.g. cyclist, bus). A similarity matrix of the dimension 689 was computed and clustering was performed to separate these into two classes. In this particular experiment the number of clusters was not automatically determined, but manually set to 2, as we focused on the evaluation of the clustering and FOD training processes. After a post processing step which takes into account the perspective deformation for each class, the number of positive training examples for the pedestrian class was 457, and 106 for the car class.

The test set contains 236 pedestrian and 357 car instances. The detection performance for both detectors is given in Figure 7. The correct detection rate is respectively about 60% and 80% for the pedestrian and car detector, for one false alarm per image. The rate is relatively low but acceptable for the pedestrian detector, and very good for the car detector, considering that neither supervision nor initial learning is required in our framework. Detection examples are shown in Figure 8.

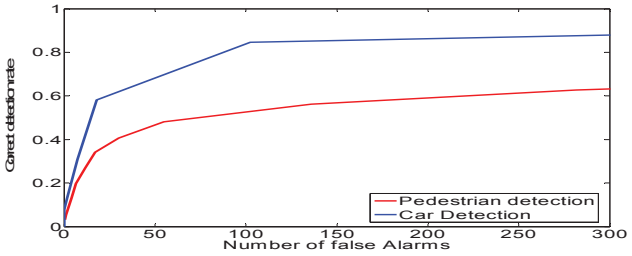

Figure 7: ROC curve for the pedestrian and car detectors.

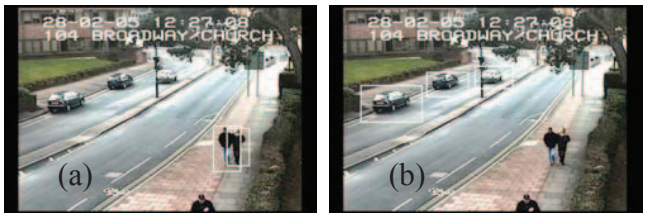

Figure 8: Some detection examples for the pedestrian detector (a) and for the car detector (b).

5. CONCLUSIONS AND FUTURE WORK

We presented here a framework that can be used to separate unlabeled object classes in an online and unsupervised fashion. The only parameter needed to be specified by the user is the number of clusters. To the best of our knowledge, our framework is the first one addressing the challenge of simultaneously training multiple object detectors in an online and unsupervised way. The features used in our approach were shown to be able to correctly discriminate and cluster multiple classes in a surveillance scenario. These classes are further used as a training set for a robust detector.

In our future research, we will focus on expanding the clustering framework to more than two classes, and on determining the number of dominant object classes in the scene in an automated fashion.

Acknowledgments: The authors wish to thank Dr. Robert P.W. Duin for productive discussions and help with implementing the clustering module.

6. REFERENCES

[1] V. Nair, J.J. Clark, "An Unsupervised, Online Learning Framework for Moving Object Detection", CVPR, 2004.

[2] A. Levin, P. Viola, Y. Freund, "Unsupervised Improvement of Visual Detectors Using Co-Training", ICCV, 2003.

[3] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection", CVPR, 2005.

[4] O. Tuzel, F. Porikli, P. Meer, "Human Detection via Classification on Riemannian Manifolds", CVPR, 2007.

[5] B. Wu, and R. Nevatia. "Improving Part based Object Detection by Unsupervised, Online Boosting", CVPR, 2007.

[6] D.G. Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, 60, 2 (2004), pp. 91-110.

[7] R. Kondor and T. Jebara, "A Kernel Between Sets of Vectors", International Conference on Machine Learning, 2003.

[8] O. Javed, S. Ali, M. Shah, "Online Detection and Classification of Moving Objects Using Progressively Improving Detectors", CVPR, 2005.

[9] H. Celik, A. Hanjalic, E.A. Hendriks, S. Boughorbel "Online training of object detectors from unlabeled surveillance video", Online Learning for Classification Workshop, CVPR, 2008.

[10] Imagery Library for Intelligent Detection Systems (i-LIDS).

[11] prtools.org: The Matlab Toolbox for Pattern Recognition.

[12] H. Celik, A. Hanjalic, E.A. Hendriks, "On the development of an autonomous and self-adaptable moving object detector", AVSS, 2007